WHAT IS CLAIMED IS:

1   1.   A method comprising,

2   in a network, encapsulating data requests generated by an

3   application in a first system;

4   transferring the encapsulated data requests to a second

5   system;

6   executing the encapsulated data requests in the second

7   system; and

8   processing in the first system responses generated by the

9   encapsulated data requests in the second system.

1   2.   The method of claim 1 in which encapsulating comprises:

2   generating an Extensible Markup Language (XML) structure

3   for each data request; and

4   converting the XML structure to an XML request.

1   3.   The method of claim 2 in which the XML structure

2   comprises a variable stream of data stored in memory of the

3   first system, the stream including an XML element for each

4   request.

1   4.   The method of claim 3 in which the XML element is a class

2   object whose data is stored to generate XML.

1   5.   The method of claim 4 in which the XML element includes

2   data from a data set object.

1 6.   The method of claim 5 in which the data set object

2 includes table dictionaries, column names and data from record

3 sets, and stored procedure parameters.

1 7.   The method of claim 1 in which transferring includes a

2 text transmission protocol.

1 8.   The method of claim 7 in which the text transmission

2 protocol is Hypertext Transfer Protocol.

1 9.   The method of claim 1 in which executing comprises:

2   de-encapsulating the encapsulated data requests by

3 parsing into request statements; and

4   executing the request statements.

1 10. The method of claim 9 further comprising:

2   translating responses from the executed request

3 statements into an XML format; and

4   sending the XML formatted responses to the first system.

1 11. A distributed application method comprising:

2   converting application requests in a first system;

3   transmitting the converted application requests to a

4 second system over a network;

5   parsing the converted application requests in the second

6 system into request statements; and

26

7   executing the request statements in the second system.

1   12.   The method of claim 11 in which converting comprises:

2   generating a data structure for storing data and

3   parameters related to an application that produced the

4   application requests;

5   translating the application requests into a standardized

6   delimited data structure stored in a memory of the first

7   system; and

8   transforming the standardized delimited data structure in

9   conjunction with the data structure into a stream of text

10   based data utilizing a Extensible Markup Language (XML)

11   format.

1   13.   The method of claim 11 in which the parsing comprises:

2   breaking down the converted application requests to an

3   executable command format utilizing data and parameters

4   related to an application.

1   14.   The method of claim 13 in which executing further

2   comprises evaluating executable commands prior to execution in

3   the second system.

1   15.   The method of claim 14 in which executing further

2   comprises evaluating results generated by the executable

3   commands.

1   16.   The method of claim 15 further comprising:

2   converting the results into a stream of text based data

3   in a standardized XML format; and

4   transmitting the converted results over the network to

5   the first system.

1   17.   An application server method comprising:

2   generating a first data structure for storing data and

3   parameters related to an application residing in the server;

4   translating application requests from the application

5   into a delimited second data structure stored in a memory;

6   generating a stream of text-based data in an Extensible

7   Markup Language (XML) format from the second data structure.

1   18.   The method of claim 17 in which the first data structure

2   includes database tables, procedure results from logic calls

3   and status/error messages.

1   19.   The method of claim 17 in which the second data structure

2   includes an element for each of the application requests.

1   20.   The method of claim 19 in which the element is a class

2   object.

1

1   21.   A method comprising:

2   in a server, receiving a stream of text-based data in an

3   Extensible Markup Language (XML) format;

4   parsing the stream into request statements; and

5   executing each of the request statements.

1   22.   The method of claim 21 in which executing further

2   comprises intercepting the request statements prior to

3   execution and applying additional logic based on a type or

4   content of the request statements.

1   23.   The method of claim 21 in which executing further

2   comprises applying additional logic to responses generated

3   from executing the request statements.

1   24.   The method of claim 21 further comprising:

2   converting responses generated from each of the executed

3   request statements into an XML format.

1   25.   A computer program product residing on a computer

2   readable medium having instructions stored thereon which, when

3   executed by the processor, cause the processor to:

4   convert application requests in a first system;

5   transmit the converted application requests to a second

6   system over a network;

7   parse the converted application requests in the second

8   system into request statements; and

9   execute the request statements in the second system.

1   26.   A computer program product residing on a computer

2   readable medium having instructions stored thereon which, when

3   executed by the processor, cause the processor to:

4   generate a first data structure for storing data and

5   parameters related to an application residing in the server;

6   translate application requests from the application into

7   a delimited second data structure stored in a memory;

8   generate a stream of text-based data in an Extensible

9   Markup Language (XML) format from the second data structure.

1   27.   A computer program product residing on a computer

2   readable medium having instructions stored thereon which, when

3   executed by the processor, cause the processor to:

4   receive a stream of text-based data in an Extensible

5   Markup Language (XML) format;

6   parse the stream into request statements; and

7   execute each of the request statements.

1   28.   An enhanced graphical user interface (GUI) method

2   comprising:

3   displaying a plurality of visual controls on an

4   input/output device; and

5   displaying at least one data enabled control on the

6   input/output device.

1   29.   The method of claim 28 in which the data enabled

2   control comprises a control having properties describing data

3   relationships to the control.

1   30.   The interface of claim 29 in which the data enabled

2   control further comprises properties describing locations of

3   data and data sources pertaining to the control.

1   31.   The method of claim 28 in which the data enabled

2   control is user-configurable.

1   32.   The method of claim 30 in which the properties

2   comprise:

3   a location of a database table;

4   a name of the database table; and

5   a column name representing the control.

1   33.   The method of claim 32 in which the properties

2   further comprise:

3   a listing of table relationships;

4   an indicator to indicate whether the control is a

5   key column in the table; and

6   an indictor to indicate whether the control is a

7   primary key column.

1   34. The method of claim 33 in which the properties

2   further comprise:

3   an indicator to indicate whether the control is part

4   of a compound primary key;

5   an indicator to indicate whether a record is locked

6   when in use; and

7   an indicator to indicate whether the control if data

8   in the control has changed.